*NASA Contractor Report 201644*

*ICASE Report No. 97-3*


# BOOLEAN ALGEBRAIC METHODS FOR PHASED-MISSION SYSTEM ANALYSIS



**Arun K. Somani**
**Kishor S. Trivedi**



*NASA Contract No. NAS1-19480*
*January 1997*

*Institute for Computer Applications in Science and Engineering*
*NASA Langley Research Center*
*Hampton, VA 23681-0001*

*Operated by Universities Space Research Association*


*National Aeronautics and Space Administration*

***Langley Research Center***
*Hampton, Virginia 23681-0001*

# Boolean Algebraic Methods for Phased-Mission System Analysis *†

Arun K. Somani
Dependable Parallel Computing and Networks Laboratory
Dept. of Electrical Eng. and Dept. of Computer Sci. and Eng.
University of Washington, Box 352500
Seattle, WA 98195-2500
email: somani@ee.washington.edu

Kishor S. Trivedi
Center for Advanced Computing and Communication
Department of Electrical and Computer Engineering
Duke University
Durham, NC, 27708-0291
email: kst@ee.duke.edu

## Abstract

Most reliability analysis techniques and tools assume that a system is used for a mission consisting of a single phase. However, multiple phases are natural in many missions. The failure rates of components, system configuration, and success (failure) criteria may vary from phase to phase. In addition, the duration of a phase may be deterministic or random. We describe a new technique for phased-mission system reliability analysis based on Boolean algebraic methods. Our technique is computationally efficient and is applicable to a large class of systems for which the failure criterion in each phase can be expressed as a fault tree (or an equivalent representation). Our technique avoids state space explosion that commonly plague Markov chain-based analysis. We develop a phase algebra to account for the effects of variable configurations and failure criteria from phase to phase. Our technique yields exact (as opposed to approximate) results. We demonstrate the use our technique by means of an example and present numerical results to show the effects of mission phases on the system reliability.

†An early version of this paper appeared in the Proceedings of ACM SIGMETRICS.
*This research in part was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681.

# 1 Introduction

The reliability analysis of ultra-reliable computer systems is an important problem for which various techniques and tools have been developed [1]-[4]. Often, reliability analysis techniques assume that the systems operate in single-phase missions. However, multiple phases are natural in many applications. The system configuration, operational requirements for individual components, the failure criteria, and the stress on the components (and thus the failure rates) may vary from phase to phase. For example, fault tolerant systems may consist of multiple subsystems employing redundancy and may have dedicated or pooled spares. A dedicated spare can replace only a single preassigned function. A pooled spare, on the other hand, has the capability of replacing any of the several functions in the system. Depending on the requirements during different phases, spares may be placed in service or removed from service to balance the system reliability and the cost of operation. The success of a redundancy management scheme determines if a system is operational or not. The usage of subsystems may also vary from phase to phase and subsystem supporting those services may remain idle or may be switched off. Furthermore, the duration of any phase may be deterministic or random. All these variations affect the system reliability.

Sometimes the effects of phased missions can be ignored in favor of simpler analysis. For example, in an airplane system, landing gear and its associated control subsystems are not required during cruising phase. So exact analysis should not ignore such failures. But, continuing to count the failure of landing gear during cruising phase has very little impact on the overall unreliability and may complicate the computation. In another example, in a space mission, several components or subsystems are used only during the take-off which is the first phase of the mission. Moreover, the failure rates of these components may be extremely high. Use of the high failure rates and the entire mission time as exposure time for all components yields inaccuratea and very high unreliability.

However, most of the time only conservative estimates are made, thus yielding the worst case unreliability of the system. One adverse effect of this is that the systems are over-designed. For economic reasons, therefore, it is desirable to perform a more accurate analysis. In particular, if one phase may see much more stress than others then it is necessary to account for these effects properly. It is not accurate to use conservative parameters for the the entire mission. On the other hand the impact of a phase with severest parameter values must not be ignored in analysis. Different aspects of phased-mission systems have been discussed by several researchers [5] - [11].


Figure 1: The three units in a system

To describe and compare the work here of others and our own, we will use a three-component system as an example. Components A, B, and C are used in a system that is employed in a mission with 3 phases. The phases are denoted as Phase X, Phase Y, and Phase Z, respectively. To show the effect of phased-mission analysis we will consider all six permutation of these three phases. That is, we will assume that the mission may go through the three phases in any order. So one particular order may be Phases X, Y, and Z or another could be Phases Z, Y, and X. The failure criteria for each of the three phases is expressed using fault trees as shown in Figure 2. In Phase X, the system fails if any of the components A, B, or C fails. In Phase Y, the system fails if component A fails or both of the components B and C fail. In Phase Z, the system fails if all three components fail. The failure rates of three components are $\lambda_a$, $\lambda_b$, and $\lambda_c$, respectively.

The corresponding (continuous time) Markov chains for all phases are shown in Figure 3. In the Markov chains, states are 3-tuples indicating up/down condition of the three components. A "1" indicates that the corresponding component is up (alive or operational) and a "0" indicates that the component has failed.

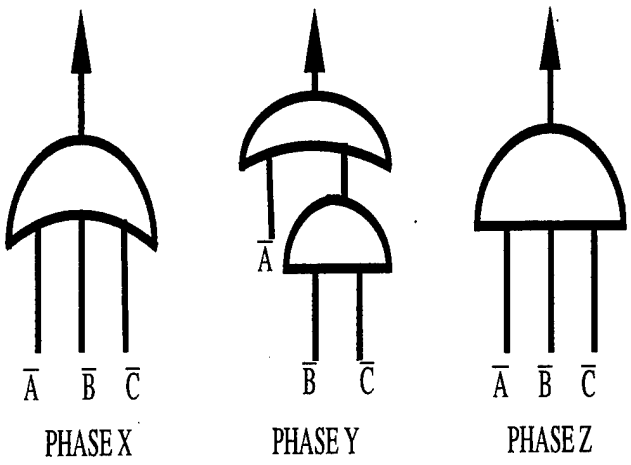



Figure 2: The failure criteria for phases expressed using fault trees

For example, a state (101) indicates that component B has failed and the other two components are up. A transition from one state to another state has a rate associated with it which is the failure rate of the component that fails. For example, a transition from state (011) to state (010) has a transition rate of $\lambda_c$. States marked $F$ are system failure states.

It should be noted that the system reliability cannot be obtained by simply solving the individual fault trees (or the corresponding Markov chains) for different phases and then appropriately manipulating them.

## 2 Related Work

Esary and Ziehms [5] (EZ approach) discuss analysis of multiple configuration systems during different phases of a mission to accomplish specified goals. In EZ approach, each phase of a system is modeled using a separate reliability block diagram (RBD). For phase $p$, each component $C$ is represented by a series of a blocks $\overline{c_1}, \overline{c_2}, \cdots, \overline{c_p}$ where $c_i$ represents the probability of failure ($\overline{c_i}$ is the probability of success) associated with component $C$ in a phase $i$ and depends on the failure rate of that component during that phase. All phase RBDs are connected in series as shown in Figure 4 for a three phase system using three components. Solution of this RBD correctly predicts the reliability of the three phase system. The problem with this approach is that a large RBD with several common events is needed, the solution of which may be computationally very expensive. Each component generates $p$ basic event for a $p$-phased system. A $k$ component system will thus have $k * p$ basic events and obtaining cut sets after accounting for common events is expensive. Approximate solution of the RBD may incur large errors. Nevertheless, for a system with "small" fault trees (in terms of number of events) and a small number of phases this method is conceptually simple. Their approach can be cast in terms of a fault-tree with repeated events and can then be solved using existing tools such as SHARPE [2].

Pedar and Sarma [6] (PS approach) carry out phased-mission analysis of aerospace computing systems using an approach similar to the EZ approach. They developed a procedure to systematically cancel out the common events in earlier phases which are accounted for in later phases. Alam and Al-Saggaf [7] (AA approach) developed a technique to analyze repairable systems in which system failure criteria and failure rates of components may vary from phase to phase.

Smotherman and Zemoudeh [9] (SZ approach) use a non-homogeneous Markov model to carry out a phased-mission system analysis. They represent the behavior of the system in each phase using a different Markov chain and each phase is represented by a separate subset of the states. The state transitions are described in terms of time dependent rates so as to include phase changes. Thus, state-dependent phase changes, random phase durations, time-varying failure and repair behavior can all be easily modeled. A

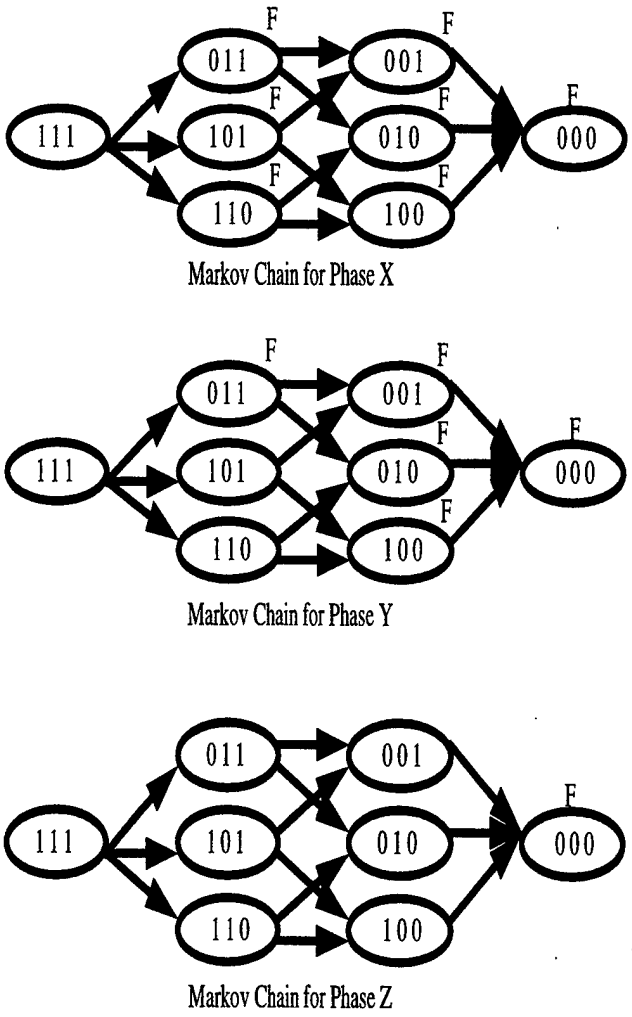



Figure 3: The Markov chains for three phases

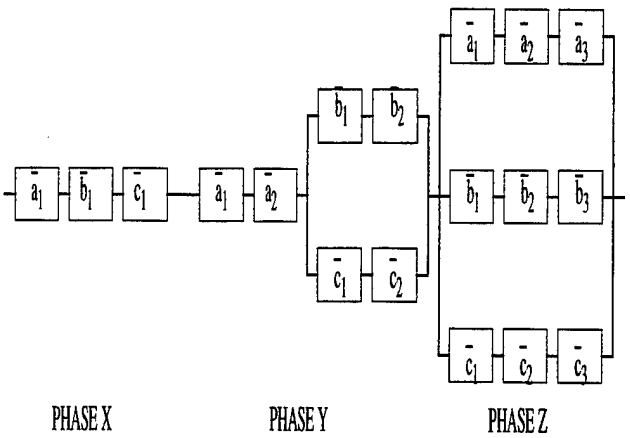



Figure 4: Reliability block diagram for a three phases system with variable configuration

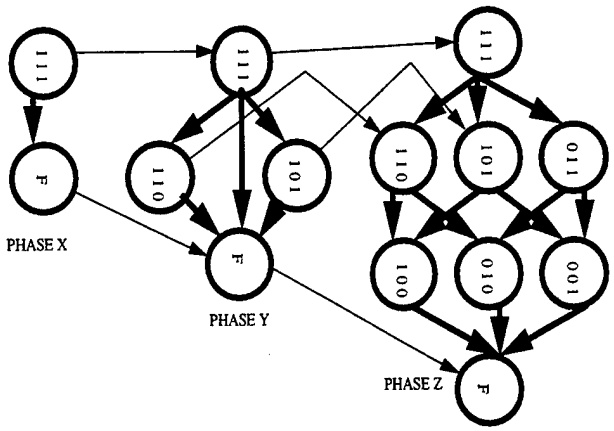


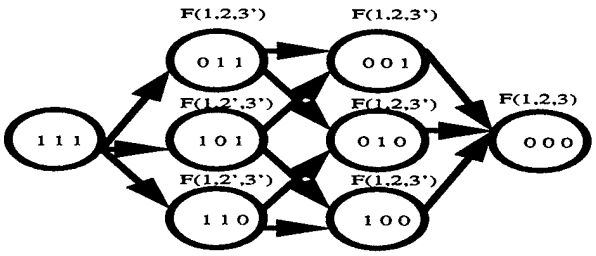

Figure 5: The multi-phase Markov chain

complete Markov chain of a three phase system of Figure 2 with phase order of X, Y, and Z is shown in Figure 5. The major drawback of this approach, like the EZ approach, is that a huge overall model is needed. The size of the state space equals the sum of the number of states in each of the individual phases. This requires large amount of storage and computation time to solve a system, thus limiting the type of system that can be analyzed.

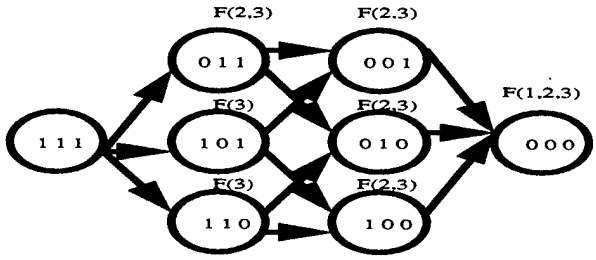
Somani et. al. [10] (SRA approach) presented a computationally efficient method to analyze multi-phased systems and a new software tool for reliability analyses of such systems. A system with variable configuration and failure criteria results in different Markov chains for different phases as shown in Figure 5. Instead of generating and solving an overall Markov chain, they advocate generating and solving separate Markov chains for individual phases. The variation in failure criteria and change in system configuration from phase to phase are accommodated by providing an efficient mapping procedure at the transition time from one phase to another. While analyzing a phase, only the states relevant to that phase, are considered. Thus each individual Markov chain is much smaller than in Smotherman and Zemoudeh [9]. For example, in Figure 5, three Markov chains with number of states 2, 4, and 8, respectively are solved instead of a single Markov chain with 12 states. Using this approach, the computation time for large systems can be reduced significantly without compromising accuracy. Phases may be of a fixed or a random duration. The reliability (or unreliability) of the system can be computed from the output of the final phase. Furthermore, the technique is sufficiently general and the most appropriate if individual phase description cannot be represented using a fault tree or RBD.

Using a similar approach, Dugan [8] (Dugan approach) suggested another method in which a single Markov chain with state space equal to the union of the state spaces of the individual phases is generated. The transition rates are parameterized with phase numbers and the Markov chain is solved $p$ times for $p$ phases. The final state occupancy probabilities of one phase become the initial state occupation probabilities for the next phase. In her approach, once a state is declared to be a system failure state in a phase, it cannot become an up state in a later phase. This could be a potential source of problem as it is possible for system to have some states that are failure states in a phase but are up states in a later phase. For example, consider the two scenarios as shown in Figure 6. In the first case (Figure 6a), phase order is Phase X, Phase Y, and Phase Z. In this case, some of the states are failure states in the first phase that are later on treated as forced failure states although they are not failure states in phases 2 and 3. Such states are marked as F(1,2',3') or F(1,2,3'). In the second case, phase order is Phase Z, Phase Y, and Phase X. In this case, there are no forced failure states.

The approach in the present paper is based on our earlier work [13]. We present a methodology (ST approach) to analyze and solve phased-mission systems in which failure rates, configuration and failure criteria can vary from phase to phase. Moreover, the failure criteria can be specified using fault trees or an




(a) Markov Chain for Phase X, Y, and Z




(b) Markov Chain for Phases Z, Y, and X

Figure 6: Two scenarios for phased-mission systems with variable configuration

equivalent representation. We believe that a majority of systems can be represented using fault trees. Our approach is similar to the EZ approach in that we do not generate any Markov chains. In contrast to the EZ approach, our method involves the solution of multiple fault trees, one per phase, similar to the SRA approach, rather than a single fault tree inclusive of all phase fault trees. The price we pay is the information we have to carry forward from phase to phase that affects the solution of the next phase's fault tree. However, solving a single large fault tree that is a combination of all phase fault trees with multiple repeated events is computationly more expensive than solving individual fault trees with some interaction. This approach has been extended by Somani [14] (Somani approach) to analyze systems which include independent repairs of components but the failure criteria can still be specified using fault trees [14]. For a given phase mission system and operating conditions, Figure 7 presents a tree of recommended approaches depending upon the problem characteristics.

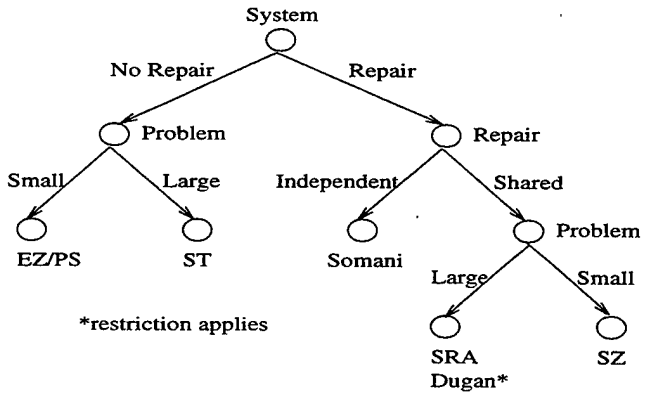



Figure 7: Recommended Approaches Tree

In the following we describe our approach in detail. First we describe some concepts which we will use throughout the paper.

# 3 Distribution Functions with Mass at Origin

One of the key concepts we will use in our method is that of cumulative distribution functions with a mass at the origin. Consider a random variable $X$ with cumulative distribution function given by

$$F_X(t) = (1 - e^{-\lambda T_1}) + e^{-\lambda T_1}(1 - e^{-\lambda t}).$$

This function has a mass at the origin given by $P(X = 0) = (1 - e^{-\lambda T_1})$ . The second term represents the continuous part of the distribution function.

In order to illustrate the use of such a CDF, consider a component with a failure rate of $\lambda$ that is used in a phased mission system. Assume that the system has just completed one phase of duration $T_1$ and is currently in the second phase. The above CDF can be assigned as the failure probability distribution of the component in the second phase. The first term in the above expression represents the probability that the component has already failed in the first phase. The second term represents the failure probability distribution for this component for the second phase. The time origin for the second phase is reinitialized to the beginning of the phase. We will use such distribution functions to represent failure probabilities of individual components during different phases.

# 4 Phased-Mission Analysis: Phase Independent Failure Criteria

In this section, we first consider a simpler scenario, a phased-mission system in which the failure criterion is phase independent. Therefore, the system configuration and the failure criteria remains unchanged from phase to phase and can be represented by the same fault tree for all phases. However, component failure rates are allowed to be phase dependent. We also assume that components behavior is independent. We first assume that phase durations are deterministic. We will relax these constraints one at a time in the following subsections.

## 4.1 Phase-Dependent Failure Rates

To account for phase-dependent failure rates, we assign a failure distribution with mass at origin to each component. Let $\lambda_{ji}$ represent the failure rate of component $j$ in phase $i$. For component $j$, the distribution function assigned in phase $k$ is given by

$$F_{C_{j,k}}(t) = (1 - e^{-\sum_{i=1}^{k-1} \lambda_{ji} T_i}) + e^{-\sum_{i=1}^{k-1} \lambda_{ji} T_i}(1 - e^{-\lambda_{jk} t}).$$

Here time $t$ is measured from the beginning of phase $k$ so that $0 \leq t \leq T_k$. $T_i$ represents the duration of phase $i$. This expression can be simplified to: $F_{C_{j,k}}(t) = 1 - e^{-\lambda_{jk} t}[e^{-\sum_{i=1}^{k-1} \lambda_{ji} T_i}]$. At the end of phase $k$, at $t = T_k$, the above expression gives the mass at the origin for phase $k+1$. A component fails during a phase only if it survives all the previous phases. The factor enclosed in square brackets above is the probability of success during first $k-1$ phases. Since the failure criteria is the same in all phases, a system fails by phase $k$ if it fails any time during the first $k$ phases. We can obtain the unreliability of the system at time $0 \leq t \leq T_k$ during phase $1 \leq k \leq p$ by evaluating the fault tree using the failure distribution function for each component as given by $F_{C_{j,k}}(t)$. Of course, if our only interest is in the failure probability for the entire mission, we evaluate the fault tree assigning a constant failure probability

$$1 - e^{-\sum_{i=1}^{i=p} \lambda_{ji} T_i},$$

to component $j$ .

It can be readily seen that the computation is correct. Since the failure rates are the only parameters that are varying from phase to phase, their impact is being considered in computing the overall failure probability

of each component. Since the fault tree is the same in all phases, a system survives only if it survives in all phases. Thus evaluating the fault tree for the last phase with the final component failure distribution functions yields the correct answer.

### 4.2 Age-Dependent Failure Rates

If the failure rates of components are phase and age dependent then we cannot count time for each phase independently. Instead, to compute the failure probability distribution, we have to account for the global (mission) time and its affect on each component. This can be achieved by assigning the failure distribution function for component $j$ in phase $k$ as follows.

$$F_{C_{j,k}}(t) = (1 - e^{-\sum_{i=1}^{k-1} \int_{CT_{i-1}}^{CT_i} \lambda_{ji}(\tau)d\tau})$$

$$+e^{-\sum_{i=1}^{k-1} \int_{CT_{i-1}}^{CT_i} \lambda_{ji}(\tau)d\tau}(1 - e^{-\int_{CT_{k-1}}^{t} \lambda_{jk}(\tau)d\tau}).$$

Here, $CT_i = \sum_{l=1}^{i} T_l$ is the sum of durations for $i$ phases and $CT_0 = 0$. The time $t$ is the cumulative time and is not reset to zero for the next phase. Time $t$ is 0 at the beginning of a mission and continues to increase through all phases. With this modification, the fault tree can be evaluated for any time $0 \leq t \leq CT_p$. The probability of failure of component $C_j$ at the end of the mission is given by

$$1 - e^{-\sum_{i=1}^{p} \int_{CT_{i-1}}^{CT_i} \lambda_{ji}(t)dt}.$$

Using $F_{C_{j,p}}(CT_p)$ as the failure probability for component $C_j$ (for all $j$), the fault tree can be evaluated to obtain the mission failure probability.

### 4.3 Random Phase Durations

To account for random phase durations, we use conditioning as above followed by the theorem of total probability. Let $F_{T_i}(t_i)$ be the distribution function for the length of phase $i$. These distributions are specified by the user. Conditioning on the durations of phases $T_1 = t_1, T_2 = t_2, \cdots, T_p = t_p$ the mission failure probability for component $j$ is given by

$$1 - e^{-\sum_{i=1}^{i=p} \lambda_{ji} t_i}.$$

Then the unconditional failure probability for component $j$ is given by

$$F_{C_{j,k}} = \int \int ..... \int [1 - e^{-\sum_{i=1}^{i=p} \lambda_{ji} t_i}] dF_{T_1}(t_1).....dF_{T_p}(t_p)$$

$$= 1 - \Pi_{i=1}^{p} F_{T_i}^{\sim}(\lambda_{ji})$$

where $F_{T_i}^{\sim}(s)$ is the LST (Laplace-Stieltejs transform) of $T_i$ and is given by $F_{T_i}^{\sim}(s) = \int_0^{\infty} e^{-st_i} dF_{T_i}(t_i)$.

This failure probability, $F_{C_{j,k}}$ can be assigned to component $C_j$ (for all $j$) and the fault tree can be evaluated to compute the unreliability of the system for the whole mission consisting of $p$ phases.

## 5 Phase-Dependent Failure Criteria

The results of the previous section apply to the cases when the failure criterion does not change from phase to phase. However, in many applications, the system configuration and the failure may change from phase to phase. There are several reasons for reconfiguration and change in failure from phase to phase. Some of these are discussed below.

1. A component is used in all phases but its operational level requirements may change. In this case, no special treatment is required for this component. The definition of operation or failed state depends on the failure criterion. This is similar to previous cases.

2. A component is used in a $n$ consecutive phases starting with some phase $k$, and is then not needed for system operation in the remaining phases. For example, some of the take-off accessories for a space mission are not needed after the take-off is completed.

3. A component is required to remain operational for some phase, is not need for the operation of a few phases and is then required again for system operation. Landing gear with its associated control mechanisms in aircrafts is a prime example of this situation.

4. Additional redundant modules are added during the operation of the system at different times.

5. Some redundant modules are removed from a subsystem operation during a mission.

6. Spare modules or operational redundant modules corresponding to one subsystem become spare or redundant modules for another subsystem.

Due to a change in failure criterion, it is possible that some combination of failures of components in one phase leads to failure of the system whereas the same combination does not lead to failure in some other phase. In Markov chain-based methods, it is easier to keep track of the system states, and therefore, change in system failure criteria could be easily accounted for. However, in the case of a fault tree, this change needs to be accounted for by considering cases when the system may fail or may not fail at the time of a phase transition. There are four possible cases which may occur at the time of a phase transition from phase $i$ to phase $i+1$.

1. A combination of component failures does not lead to system failure in both phases $i$ and $i+1$.

2. A combination of component failures leads to system failure in both phases $i$ and $i+1$.

3. A combination of component failures does not lead to system failure in phase $i$ but leads to system failure in phase $i+1$.

4. A combination of component failures leads to system failure in phase $i$ but not in phase $i+1$.

The first two cases require treatment similar to that in the previous section as the failure criteria does not change from phase $i$ to phase $i+1$ with respect to the failure combination under consideration. Failure combinations in the third case above should be treated as failures in the earlier phase $i$ as well. This is because such combinations, once present during a phase are bound to lead to the system failure eventually at the transition time when the system enters this later phase. These are referred to as latent failures in [11]. Hence a more stringent criterion should be applied with respect to these combinations. So we can assume that all failure combinations in phase $i+1$ are also failure combinations in phase $i$ (but not vice versa). Hence for the first three cases, the unreliability can be evaluated by evaluating the fault tree for the last phase using the approach of Section 4.

The failure combinations which imply system failure in phase $i$, but do not lead to system failure in subsequent phases, as is the fourth case, should be handled more carefully. We need to account for the probability of occurrence of these failure combinations until phase $i$. Any probability attributed to such combinations of component failures in later phases does not contribute towards system unreliability. Esary and Ziehms account for this by cascading the phase reliability blocks. However, as mentioned earlier, that leads to a more expensive computation. We present our method of handling such failure combinations below.

Our methodology consists of the following steps. We divide the system unreliability of a phased mission system into two parts: (i) common failure combinations; and (ii) phase failure combinations. We evaluate the unreliability due to these two parts using the following procedure.

## 5.1 Common Failure Combinations

The first part, common failure combinations, includes the probability of those component failure combinations which are common to all phases after the most stringent criterion has been applied to all phases. That is, if a combination leads to system failure in phase $i+1$, then it is considered a failure combination in phase $i$ as well. Thus the common failure combinations essentially include the failure combination specified for the last phase.

The unreliability due to common failure combinations can be computed using the method described in the previous section for analyzing phased-mission system with phase-independent failure criteria. That is, we compute the failure probability distribution for individual component and then evaluate the common fault tree which is the fault tree for the last phase.

## 5.2 Phase Failure Combinations

The second part, phase failure combinations, includes the probability of all failures specific to individual phases after applying the most stringent success criterion in each phase. For phase $i$, this part include the probability of only those component failure combinations which contribute to system failure in phase $i$ but are considered operational in all subsequent phases.

Unreliability due to the second part requires additional computations. For each phase, we need to identify and compute the probability of component failure combinations which lead to system failure in that phase and does not imply system failure in any subsequent phase. Let $E_i$ be the Boolean logic expression specifying the failure combinations for phase $i$. Then phase failure combinations for phase $i$ ($PFC_i$), which are treated as success combinations for the all subsequent phases are given by

$$PFC_i = (\cdots((E_i \wedge \overline{E_{i+1}}) \wedge \overline{E_{i+2}}) \cdots \wedge \overline{E_p}).$$

In the above expression, we include only those combinations which are failure combinations in phase $i$ but are not failure combinations in any of the subsequent phases. This expression can be simplified as

$$PFC_i = E_i \wedge \overline{(E_{i+1} \vee \cdots \vee E_p)}.$$

## 5.3 Phase Algebra

Let $\overline{A} = 1$ mean that component $A$ has failed. Then $A = 0$ implies that component $A$ has failed and $A = 1$ means that component $A$ is operational. Using this notation, for the system described in Figure 2 the following Boolean expressions describe the failure combinations for phases X, Y, and Z.

$$E_X = \overline{A} + \overline{B} + \overline{C}$$
$$E_Y = \overline{A} + \overline{B}\,\overline{C}$$
$$E_Z = \overline{A}\,\overline{B}\,\overline{C}$$

It should be noted that in the expression for $PFC_i$, event $\overline{A}$ denotes the failure of component $A$ in phase $i$ only. Thus for each phase, we need to define a separate symbol for each component. This is very similar to Esary and Ziehms notation where they have a separate symbol denoting failure of a component in each phase. Let $A_i = 1$ denote the event that component $A$ is operational during the interval from the start of the mission until the end of phase $i$. This automatically implies that the component is operational during all earlier phases as well. With this addition, the Boolean expressions for phases X, Y, and Z used in system phase $i$ are denoted by $E_{iX}$, $E_{iY}$, and $E_{iZ}$, respectively, and are given by the following.

$$E_{iX} = \overline{A_i} + \overline{B_i} + \overline{C_i}$$
$$E_{iY} = \overline{A_i} + \overline{B_i}\,\overline{C_i}$$

$E_{iZ} = \overline{A_i}\ \overline{B_i}\ \overline{C_i}$

When the expression for $PFC_i$ is simplified, we need to merge different combinations of such terms. This could be a little tricky and needs special treatment. Let $\alpha$ and $\beta$ be two phases and let $\alpha < \beta$. The rules in Table 1 can be used to simplify the logic expressions.

Table 1: Combining rules

| | | | | | |
|---|---|---|---|---|---|
| $A_\alpha\ A_\beta$ | $\rightarrow$ | $A_\beta$ | $\overline{A_\alpha} + \overline{A_\beta}$ | $\rightarrow$ | $\overline{A_\beta}$ |
| $\overline{A_\alpha}\ \overline{A_\beta}$ | $\rightarrow$ | $\overline{A_\alpha}$ | $A_\alpha + A_\beta$ | $\rightarrow$ | $A_\alpha$ |
| $\overline{A_\alpha}\ A_\beta$ | $\rightarrow$ | 0 | $A_\alpha + \overline{A_\beta}$ | $\rightarrow$ | 1 |

$A_\alpha\overline{A_\beta}$ and $\overline{A_\alpha} + A_\beta$ do not simplify any further. What the first combination means is that component $A$ is operational until the end of phase $\alpha$ and then fails sometime between the end of phase $\alpha$ and end of phase $\beta$. The second term has no physical meaning. Also, if a component fails during a phase and then it is required to be operational during a later phase, then the two events cannot be satisfied at the same time. That is why $\overline{A_\alpha}A_\beta \rightarrow 0$ holds.

The correctness of these relations can be verified by considering the following. Let $a_\alpha = 1$ denote that the component $A$ is operational during phase $\alpha$ only. Then $A_\alpha = a_1 a_2 \cdots a_\alpha$ and $A_\beta = a_1 a_2 \cdots a_\beta$. Now by substituting these values on both sides of each of these relations, we can verify that the relations hold.

## 5.4 System Unreliability

Using the phase algebra to compute the system unreliability, we first compute the $PFC_i$'s for all phases. Then the system unreliability is given by

$$UR = P(E_p) + \sum_{i=1}^{p-1} P(PFC_i)$$

where $P(E_p)$ is the probability of failure evaluated using the fault tree $E_p$ of phase $p$ (the last phase) and the failure distribution function calculated for each component as described in Section 3. $P(PFC_i)$ is the probability of phase failure combinations for phase $i$. To calculate $PFC_i$'s, we will require probability of events such as a component remains operational during all phases starting from 1 to $i$, or a component remains operational during phase 1 to phase $k$ and then fails during phase $k+1$ to phase $i$ for some $k$. Such probabilities can also be calculated using the techniques defined in Section 3.

## 5.5 Example

In this section, we demonstrate our technique using the example described in Figure 2. This system has three components and we describe three phases, X, Y, and Z. To show the difference between various techniques, we will consider all the six permutations of three phases. The failure combinations of three phases are defined by $E_X$, $E_Y$, and $E_Z$ above.

We discuss each of the six permutations below.

**Permutation X Y Z.** In this case first phase is phase X, followed by phase Y, which is then followed by phase Z. So the $PFC_i$ functions are obtained as follows.

$$
\begin{aligned}
PFC_1 &= (E_{1X} \cdot \overline{E_{2Y}}) \cdot \overline{E_{3Z}} \\
&= ((\overline{A_1} + \overline{B_1} + \overline{C_1}).(\overline{A_2 + \overline{B_2}\ \overline{C_2}})).(\overline{A_3}\ \overline{B_3}\ \overline{C_3}) \\
&= A_3B_2\overline{C_1} + A_3\overline{B_1}C_2 + A_2B_3\overline{C_1} + A_2\overline{B_1}C_3
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
PFC_2 &= E_{2Y}.\overline{E_{3Z}} \\
&= (\overline{A_2} + \overline{B_2}\ \overline{C_2}).(\overline{A_3}\ \overline{B_3}\ \overline{C_3}) \\
&= A_3\overline{B_2}\ \overline{C_2} + \overline{A_2}B_3 + \overline{A_2}C_3
\end{aligned} \tag{2}
$$

Then the system unreliability is given by

$$
UR_{XYZ} = P(E_{3Z}) + P(PFC_1) + P(PFC_2). \tag{3}
$$

In the equation above

$$
P(E_{3Z}) = P(\overline{A_3}) \cdot P(\overline{B_3}) \cdot P(\overline{C_3}). \tag{4}
$$

The other two summands of Equation 3 are computed as follows.

$$
\begin{aligned}
P(PFC_1) &= P(A_3B_2\overline{C_1} + A_3\overline{B_1}C_2 + A_2B_3\overline{C_1} + A_2\overline{B_1}C_3) + A_2\overline{B_1}C_3) \cdot (\overline{A_3} + \overline{B_2} + C_1)) \\
&= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2 + A_2\overline{A_3}B_3\overline{C_1} + A_2\overline{B_1}C_3) \\
&= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2) + P((A_2\overline{A_3}B_3\overline{C_1} + A_2\overline{B_1}C_3) \cdot (\overline{A_3} + B_1 + \overline{C_2})) \\
&= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2) + P(A_2\overline{A_3}B_3\overline{C_1} + A_2\overline{A_3}\ \overline{B_1}C_3) \\
&= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2) + P(A_2\overline{A_3}B_3\overline{C_1}) + P((A_2\overline{A_3}\ \overline{B_1}C_3) \cdot (\overline{A_2} + A_3 + \overline{B_3} + C_1)) \\
&= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2) + P(A_2\overline{A_3}B_3\overline{C_1}) + P(A_2\overline{A_3}\ \overline{B_1}C_3)
\end{aligned}
$$

*and*

$$
\begin{aligned}
P(PFC_2) &= P(A_3\overline{B_2}\ \overline{C_2} + \overline{A_2}B_3 + \overline{A_2}C_3) \\
&= P(A_3\overline{B_2}\ \overline{C_2}) + P((\overline{A_2}B_3 + \overline{A_2}C_3) \cdot (\overline{A_3} + B_2 + C_2)) \\
&= P(A_3\overline{B_2}\ \overline{C_2}) + P(\overline{A_2}B_3 + \overline{A_2}C_3) \\
&= P(A_3\overline{B_2}\ \overline{C_2}) + P(\overline{A_2}B_3) + P((\overline{A_2}C_3) \cdot (A_2 + \overline{B_3})) \\
&= P(A_3\overline{B_2}\ \overline{C_2}) + P(\overline{A_2}B_3) + P(\overline{A_2}B_3C_3)
\end{aligned} \tag{5}
$$

It is easy to compute the probability of failure in phase 3 using the failure distributions for individual components. Any fault tree solver such as SHARPE [2] can be used to compute it. Similarly, the probability of expressions in Equation 2 can be evaluated after simplifying the expressions as a sum of disjoint products using algorithm such as the one described in [12] and depicted in Equation 5.

**Permutation X Z Y.** In this case the first phase is phase X, followed by phase Z, followed by phase Y. Without going too much in details, the $PFC_i$ functions are computed as follows.

$$
PFC_1 = (E_{1X} \cdot \overline{E_{2Z}}) \cdot \overline{E_{3Y}} = A_3B_3\overline{C_1} + A_3\overline{B_1}C_3
$$

and

$$
PFC_2 = E_{2Z}.\overline{E_{3Y}} = \phi
$$

The last phase in this case is phase Y. The system unreliability can be computed using

$$
\begin{aligned}
UR_{XZY} &= P(E_{3Y}) + P(PFC_1) + P(PFC_2) \\
&= P(\overline{A_3}) + P(A_3\overline{B_3}\ \overline{C_3}) + P(A_3B_3\overline{C_1}) + P(A_3\overline{B_1}C_3).
\end{aligned}
$$

**Permutation Y X Z.** In this case, the $PFC_i$ functions are computed as follows.

$$PFC_1 = (E_{1Y} \cdot \overline{E_{2X}}) \cdot \overline{E_{3Z}} = \phi$$

and

$$PFC_2 = E_{2X}.\overline{E_{3Z}} = A_3(\overline{B_2} + \overline{C_2}) + B_3(\overline{A_2} + \overline{C_2}) + C_3(\overline{A_2} + \overline{B_2})$$

The last phase in this case is phase Z. The system unreliability can be computed using the following. (We are omitting details of simplification.)

$$\begin{aligned} UR_{YXZ} &= P(E_{3Z}) + P(PFC_1) + P(PFC_2) \\ &= P(\overline{A_3}) \cdot P(\overline{B_3}) \cdot P(\overline{C_3}) + P(A_3\overline{B_2}) + P(A_3B_2\overline{C_2}) + P(\overline{A_2}B_3) + P(\overline{A_2}\ \overline{B_3}C_3) \\ &+P(A_2\overline{A_3}B_3\overline{C_2}) + P(A_2\overline{A_3}\ \overline{B_2}C_3) \end{aligned}$$

**Permutation Y Z X.** In this case, the $PFC_i$ functions are computed as follows.

$$PFC_1 = (E_{1Y} \cdot \overline{E_{2Z}}) \cdot \overline{E_{3X}} = \phi$$

and

$$PFC_2 = E_{2Z}.\overline{E_{3X}} = \phi$$

The last phase in this case is phase X. The system unreliability can be computed using the following.

$$\begin{aligned} UR_{YZX} &= P(E_{3X}) + P(PFC_1) + P(PFC_2) \\ &= P(\overline{A_3}) + P(\overline{B_3}) + P(\overline{C_3}) \end{aligned}$$

**Permutation Z X Y.** In this case, the $PFC_i$ functions are computed as follows.

$$PFC_1 = (E_{1Z} \cdot \overline{E_{2X}}) \cdot \overline{E_{3Y}} = \phi$$

and

$$PFC_2 = E_{2X}.\overline{E_{3Y}} = A_3B_3\overline{C_2} + A_3\overline{B_2}C_3$$

The last phase in this case is phase Y. The system unreliability can be computed using the following.

$$\begin{aligned} UR_{ZXY} &= P(E_{3Y}) + P(PFC_1) + P(PFC_2) \\ &= P(\overline{A_3}) + P(A_3\overline{B_3}\ \overline{C_3}) + P(A_3B_3\overline{C_2}) + P(A_3\overline{B_2}C_3) \end{aligned}$$

**Permutation Z Y X.** In this case, the $PFC_i$ functions are computed as follows.

$$PFC_1 = (E_{1Z} \cdot \overline{E_{2Y}}) \cdot \overline{E_{3X}} = \phi$$

and

$$PFC_2 = E_{2Y}.\overline{E_{3X}} = \phi$$

The last phase in this case is phase X. The system unreliability can be computed using the following.

$$\begin{aligned} UR_{YZX} &= P(E_{3X}) + P(PFC_1) + P(PFC_2) \\ &= P(\overline{A_3}) + P(\overline{B_3}) + P(\overline{C_3}) \end{aligned}$$

## 5.6 Exact Solution Using Markov Chains

The same three-component system can be analyzed using Markov chains for the six permutations of phases. There are eight possible states in each phase as depicted in Figure 3. Using the same notation for the names of states, i.e., state 101 represents that components A and C are operational and component B has failed, we can derive expressions for states occupancy probabilities (SOPs) at the end of each phase. Depending

on the failure criteria, for the failure states in phase $i$, the initial state occupancy probability for the same state in phase $i+1$ is zero.

Let $P_{i(s)}$ denote the SOP for phase $i$ of state $s$ where $s \in \{000, 001, 010, 011, 100, 101, 110, 111\}$ and $i = 1,\ 2$, and 3. Again, let $T_i$ denote the phase duration for phase $i$ and let $CT_i$ denote the sum of durations of first $i$ phases. Let $\lambda_{A_i}$, be the failure rates of components A, B, and C, respectively, in phase $i$. Using these notations, the SOPs for phase $i$ can be derived using the SOPs for phase $i-1$ and are given in Equation 6.

Using the relationship in Equation 6, we can compute the SOPs for operational states for each phase. The unreliability at the end of each phase is given by 1 - sum of SOPs of operational states in that phase. At the end of that phase, SOP for the failure states in that phase can be set to zero as this probability mass is not carried forward to the next phase to success states. For example, for the case of permutation X Y Z, initially $P_{0(s)} = 0.0$ for all states where $s \neq 111$ and $P_{0(111)} = 1.0$. Using these values and the failure criteria for phase X, at the end of phase X, we assign $P_{1(s)}(CT_1) = 0.0$ for all states where $s \neq 111$ and $P_{1(111)}(CT_1)$ is calculated Equation 6. Using, these values and the failure criteria of phase Y, we can compute SOPs for phase 2. At the beginning of phase 3, we assign $P_{2(s)}(CT_2) = 0.0$ where $s \in \{000, 001, 010, 011, 100\}$ and compute $P_{2(s)}(CT_2) = 0.0$ where $s \in \{101, 110, 111\}$ using relations defined in Equation 6. Finally, using these results of phase 2, we can calculate $P_{3(s)}(CT_3)$ where $s \in \{001, 010, 011, 100, 101, 110, 111\}$.

Sometimes a backwards or need-based computation may be more useful. For example, for permutation Z Y X, we only need to calculate $P_{3(111)}(CT_3)$ which requires only $P_{2(111)}(CT_2)$. This, in turn, requires the computation of $P_{1(111)}(CT_1)$ which can be calculated using $P_{0(111)}(CT_0) = 1.0$. Finally, the unreliability for the three phase system is $1 - P_{3(111)}(CT_3)$. However, intermediate unreliabilities at the end of phases 1 and 2 may require more computation.

$$
\begin{aligned}
P_{i(111)}(CT_{i-1}+t) &= P_{i-1(111)}(CT_{i-1}) * e^{-\lambda_{A_i}t} * e^{-\lambda_{B_i}t} * e^{-\lambda_{C_i}t} \\
P_{i(110)}(CT_{i-1}+t) &= P_{i-1(111)}(CT_{i-1}) * e^{-\lambda_{A_i}t} * e^{-\lambda_{B_i}t} * (1 - e^{-\lambda_{C_i}t}) \\
&\quad + P_{i-1(110)}(CT_{i-1}) * e^{-\lambda_{A_i}t} * e^{-\lambda_{B_i}t} \\
P_{i(101)}(CT_{i-1}+t) &= P_{i-1(111)}(CT_{i-1}) * e^{-\lambda_{A_i}t} * (1 - e^{-\lambda_{B_i}t}) * e^{-\lambda_{C_i}t} \\
&\quad + P_{i-1(101)}(CT_{i-1}) * e^{-\lambda_{A_i}t} * e^{-\lambda_{C_i}t} \\
P_{i(011)}(CT_{i-1}+t) &= P_{i-1(111)}(CT_{i-1}) * e^{-\lambda_{C_i}t} * (1 - e^{-\lambda_{A_i}t}) * e^{-\lambda_{B_i}t} \\
&\quad + P_{i-1(011)}(CT_{i-1}) * e^{-\lambda_{B_i}t} * e^{-\lambda_{C_i}t} \\
P_{i(100)}(CT_{i-1}+t) &= P_{i-1(111)}(CT_{i-1}) * e^{-\lambda_{A_i}t} * (1 - e^{-\lambda_{B_i}t}) * (1 - e^{-\lambda_{C_i}t}) \\
&\quad + P_{i-1(100)}(CT_{i-1}) * e^{-\lambda_{A_i}t} + P_{i-1(110)}(CT_{i-1}) * e^{-\lambda_{A_i}t} * (1 - e^{-\lambda_{B_i}t}) \\
&\quad + P_{i-1(101)}(CT_{i-1}) * e^{-\lambda_{A_i}t} * (1 - e^{-\lambda_{C_i}t}) \\
P_{i(010)}(CT_{i-1}+t) &= P_{i-1(111)}(CT_{i-1}) * e^{-\lambda_{B_i}t} * (1 - e^{-\lambda_{A_i}t}) * (1 - e^{-\lambda_{C_i}t}) \\
&\quad + P_{i-1(010)}(CT_{i-1}) * e^{-\lambda_{B_i}t} + P_{i-1(110)}(CT_{i-1}) * e^{-\lambda_{B_i}t} * (1 - e^{-\lambda_{A_i}t}) \\
&\quad + P_{i-1(011)}(CT_{i-1}) * e^{-\lambda_{B_i}t} * (1 - e^{-\lambda_{C_i}t}) \\
P_{i(001)}(CT_{i-1}+t) &= P_{i-1(111)}(CT_{i-1}) * e^{-\lambda_{C_i}t} * (1 - e^{-\lambda_{A_i}t}) * (1 - e^{-\lambda_{B_i}t}) \\
&\quad + P_{i-1(001)}(CT_{i-1}) * e^{-\lambda_{C_i}t} + P_{i-1(101)}(CT_{i-1}) * (1 - e^{-\lambda_{A_i}t}) * e^{-\lambda_{C_i}t} \\
&\quad + P_{i-1(011)}(CT_{i-1}) * (1 - e^{-\lambda_{B_i}t}) * e^{-\lambda_{C_i}t}
\end{aligned}
\tag{6}
$$

## 5.7 Comparison of Accurate and Worst Case Scenarios

We analyze the above six scenarios using the technique discussed in this paper using Esary and Ziehms approach, analytic solution of Markov chains, phased-mission approach of [10] and [9], and the phased-mission approach of [8]. We assume that the durations of all the three phases are 10 hours each and the failure rate of each of the components is 0.0001/hour. Thus the input data do not skew results in any direction as all components are similar and all phases are similar. The results are only affected by the sequencing of phases and system failure criteria.

We obtain the results shown in Tables 2 and 3. The results for the six permutations of phases X, Y, and Z, are obtained (and listed) at the end of each phase. When the worst case criterion is applied, that is a failed state in one phase is considered as failed state in all subsequent phases, the results for unreliability

Table 2: Accurate Analysis: $T_1 = T_2 = T_3 = 10$ hours

| Perm | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| X Y Z | 0.002995504 | 0.003993006 | 0.003993009 |
| X Z Y | 0.002995504 | 0.002995505 | 0.004991493 |
| Y X Z | 0.001000498 | 0.005982036 | 0.005982037 |
| Y Z X | 0.001000498 | 0.001000502 | 0.008959621 |
| Z X Y | 0.000000001 | 0.005982036 | 0.006976549 |
| Z Y X | 0.000000001 | 0.002001985 | 0.008959621 |

Table 3: Worst Case Analysis: $T_1 = T_2 = T_3 = 10$ hours

| Perm | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| X Y Z | 0.002995504 | 0.005982036 | 0.008959621 |
| X Z Y | 0.002995504 | 0.005982036 | 0.008959621 |
| Y X Z | 0.001000498 | 0.005982036 | 0.008959621 |
| Y Z X | 0.001000498 | 0.002001985 | 0.008959621 |
| Z X Y | 0.000000001 | 0.005982036 | 0.008959621 |
| Z Y X | 0.000000001 | 0.002001985 | 0.008959621 |

can be very high. That is the reason the result in the first row in Tables 2 and 3 differ so much in each case.

The important thing to observe here is that when we allow failure combinations (failure states in Markov chains) to become operational combinations (up states in Markov chains) in a later phase, then the overall unreliability of the system could be substantially lower as is the case in the last column. For example, in a spacecraft, launch is the most important activity. After that, all launch related activities or components which could have caused failure during launch will not make any further difference. Thus those failure combinations are operational combinations for the rest of the mission.

To further explore the impact of phase configurations and durations of phases, we varied the phase durations. In the first variation, we assume that the first phase is always of 1 hour duration, the second phase is of 10 hour duration, and the third phase is of 100 hour duration irrespective of the types of phase configurations, X, Y, or Z, used during these phases. The results for this variation for the two cases are shown in Tables 4 and 5, respectively. In another variation, we assume that the phase X is always of 1 hour duration, phase Y is always of 10 hours duration, and phase Z is always of 100 hours duration irrespective of where in the mission these phase configurations are used. The results are given in Tables 6 and 7, respectively. In this case, the results differ by more than an order of magnitude depending on the ordering of the phases. If the strongest success criteria is during the beginning of phases, then phased-mission analysis is more meaningful.

It should be noted that the techniques in [10], [8], and [9] are capable of handling the more general case of repairable systems while the technique discussed by Esary and Ziehms as well as the one presented in this paper are both restricted to the cases of non-repairable systems. The technique in [9] is the most general but also the most expensive in computation time and in this case will yield the same result as in [10] because neither of these make any approximations.

Table 4: Accurate: $T_1 = 1$, $T_2 = 10$, $T_3 = 10$ hours

| Perm | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| X Y Z | 0.000299955 | 0.001300153 | 0.001301332 |
| X Z Y | 0.000299955 | 0.000299956 | 0.011354728 |
| Y X Z | 0.000100005 | 0.003294561 | 0.003295543 |
| Y Z X | 0.000100005 | 0.000100006 | 0.032751658 |
| Z X Y | 0.000000000 | 0.003294561 | 0.013309644 |
| Z Y X | 0.000000000 | 0.001100603 | 0.032751658 |

Table 5: Worst Case: $T_1 = 1$, $T_2 = 10$, $T_3 = 10$ hours

| Perm | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| X Y Z | 0.000299955 | 0.003294561 | 0.032751658 |
| X Z Y | 0.000299955 | 0.003294561 | 0.032751658 |
| Y X Z | 0.000100005 | 0.003294561 | 0.032751658 |
| Y Z X | 0.000100005 | 0.000200020 | 0.032751658 |
| Z X Y | 0.000000000 | 0.003294561 | 0.032751658 |
| Z Y X | 0.000000000 | 0.001100603 | 0.032751658 |

Table 6: Accurate: $T_X = 1$, $T_Y = 10$, $T_Z = 10$ hours

| Perm | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| X Y Z | 0.000299955 | 0.001300153 | 0.001301332 |
| X Z Y | 0.000299955 | 0.000300940 | 0.011354728 |
| Y X Z | 0.001000498 | 0.003294561 | 0.003295543 |
| Y Z X | 0.001000498 | 0.001001678 | 0.032751658 |
| Z X Y | 0.000000985 | 0.029845556 | 0.030816194 |
| Z Y X | 0.000000985 | 0.011058089 | 0.032751658 |

Table 7: Worst Case: $T_X = 1$, $T_Y = 10$, $T_Z = 10$ hours

| Perm | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| X Y Z | 0.000299955 | 0.003294561 | 0.032751658 |
| X Z Y | 0.000299955 | 0.003294561 | 0.032751658 |
| Y X Z | 0.001000498 | 0.003294561 | 0.032751658 |
| Y Z X | 0.001000498 | 0.011058089 | 0.032751658 |
| Z X Y | 0.000000985 | 0.029845556 | 0.032751658 |
| Z Y X | 0.000000985 | 0.011058089 | 0.032751658 |

# 6 Computing Transient Behavior

In the previous section, we computed unreliability at the end of a mission, that is, the end of the last phase. Sometime one may be interested in computing the unreliability behavior during all phases. This means we need to compute unreliability for each phase as a function of time. It turns out that this is not at all expensive as $PFC$ calculation is recursive and it easily accommodates this computation.

Recall that $PFC$ for a phase is computed as

$$PFC_i = E_i \wedge \overline{(E_{i+1} \vee \cdots \vee E_p)}.$$

Also, the unreliability at the end of a mission is computed using the expression

$$UR = P(E_p) + \sum_{i=1}^{p-1} P(PFC_i).$$

If in a $p$ phase system, we define $PFC_p = E_p$ then unreliability for a $p$ phase system can be written as

$$UR = \sum_{i=1}^{p} P(PFC_i).$$

Thus, to compute reliability at the end of phase $k$, we need $PFC_1$, $PFC_2$, $\cdots$, $PFC_k$ where the $PFCs$ must be calculated using phase $k$ as the last phase. We define $PFC_{i,k}$ as the $PFC$ of phase $i$, $i < k$, assuming phase $k$ as the last phase. Then the following relation holds.

$$PFC_{i,k} = PFC_{i,k-1} \wedge \overline{E_k}$$

The unreliability of the $k$ phase is computed by using the following relation.

$$UR_k = \sum_{i=1}^{k} P(PFC_{i,k})$$

and the $PFC_{i,k}$ can be computed recursively using the results of $PFC_{i,k-1}$ and $E_k$. With this recursive relation, one may compute reliability of phase $k$ using the result of phase $k-1$.

It should also be noticed that at the transition of a phase, one may see a sudden jump in unreliability. This happens if the next phase has more stringent success criteria than the current phase. We define this as latent failure as the system may fail as soon as the phase change occurs. For example, when an aircraft is flying, the landing gear is not important. However, as soon as the landing phase begins, and if the landing gear has failed, the system will fail. To compute unreliability increase due to phase change from phase $i$ to phase $i+1$, one must compute $UR_i$. Then, one must assume a phase $i+1$ with failure criteria of defined by

Table 8: Transient Analysis: $T_1 = T_2 = T_3 = 10$ hours

| | Phase Order Z, Y, X | | Phase Order X, Y, Z | |
|---|---|---|---|---|
| Time | UR | Latent UR | UR | Latent UR |
| 00+ Hours | 0.000000000 | N/A | 0.000000000 | N/A |
| 10- Hours | 0.000000001 | 0.001000497 | 0.002995505 | 0.000000000 |
| 10+ Hours | 0.001000498 | N/A | 0.002995505 | N/A |
| 20- Hours | 0.002001985 | 0.004980548 | 0.003990057 | 0.000000000 |
| 20+ Hours | 0.006982533 | N/A | 0.003990057 | N/A |
| 30- Hours | 0.008959621 | 0.000000000 | 0.003993009 | 0.000000000 |

$E_{i+1}$ but with phase duration of time $T_{i+1} \approx 0$. Thus no actual failure in phase $i+1$ contributes towards unreliability. however, all latent failures are accounted for as the failure criteria of phase $i+1$ is used to determine $PFCs$.

For example, consider the same example with Phases Z, Y, and X as the first, second, and third phases, respectively. In this case the $PFCs$ for each phase assuming the first, second, and the third phase as the last phase are given by:

$$\begin{aligned}
PFC_{11} &= E_{1Z} \\
&= \overline{A_1}\ \overline{B_1}\ \overline{C_1} \\
PFC_{12} &= PFC_{11} \wedge \overline{E_{2Y}} \\
&= (\overline{A_1}\ \overline{B_1}\ \overline{C_1}).(\overline{\overline{A_2} + \overline{B_2}\ \overline{C_2}}) \\
&= A_2\ B_2\ \overline{C_1} + A_2\ \overline{B_1}\ C_2 \\
PFC_{22} &= E_{2Y} \\
&= \overline{A_2} + \overline{B_2}\ \overline{C_2} \\
PFC_{13} &= PFC_{12} \wedge \overline{E_{3X}} \\
&= (A_2\ B_2\ \overline{C_1} + A_2\ \overline{B_1}\ C_2).(\overline{\overline{A_3}\ \overline{B_3}\ \overline{C_3}}) \\
&= \phi \\
PFC_{23} &= PFC_{22} \wedge \overline{E_{3X}} \\
&= (\overline{A_2} + \overline{B_2}\ \overline{C_2}).(\overline{\overline{A_3}\ \overline{B_3}\ \overline{C_3}}) \\
&= \phi \\
PFC_{33} &= E_{3X} \\
&= \overline{A_1} + \overline{B_1} + \overline{C_1}
\end{aligned} \tag{7}$$

Using the above equations, we can compute the unreliability for each phase. For example, in the above scenario, for the durations of all the three phases being 10 hours each and the failure rate of each of the component being 0.0001/hours, we obtain reliability at the beginning and end of each phase as given in Table 8. As can be seen, at each transition of a phase, we see a jump in unreliability which is essentially due to a more stringent failure criteria. On the other hand, if the failure criteria were to be more relaxed, as will be the case for phase order X, Y, and Z, there is no latent failure as shown in Table 8.

Using the above method, the best and the worst case reliability values for all combination of phases are shown in Figures 8 and 9, respectively. It is easy to see the amount of error one may accumulate if proper care is not taken in the computation.

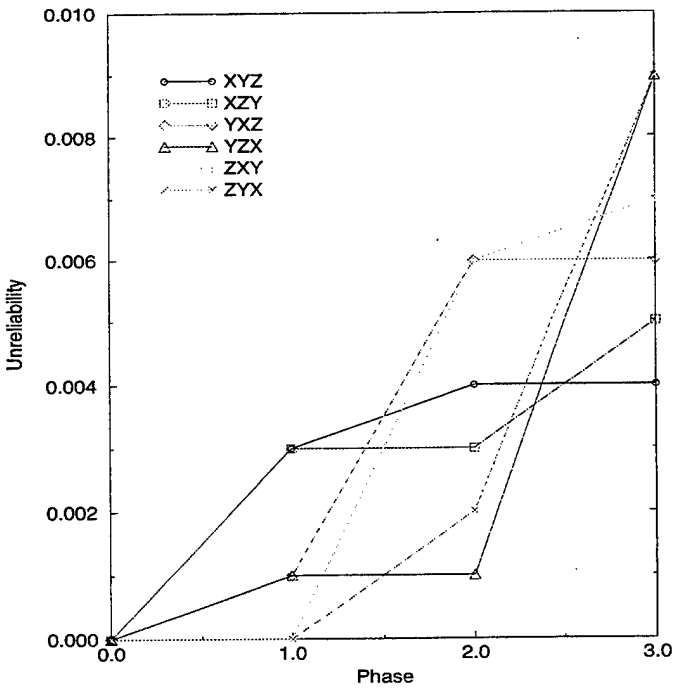



Figure 8: The best case reliability values for six combinations

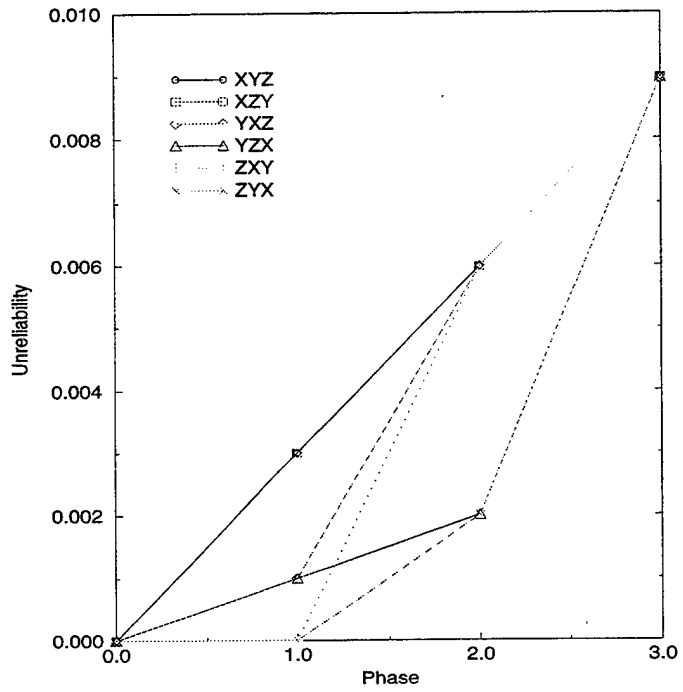



Figure 9: The worst case reliability values for six combinations

# 7 Conclusions

We have presented a technique to analyze phased-mission systems using fault trees. This technique yields accurate results and is simpler in concept and computation. For this purpose, we develop a phase algebra that allows us to efficiently compute the probability of all possible combinations contributing to failure in phased-mission systems during individual phases. This technique will be very useful for a large class of systems where the system behavior can be described using fault trees. Currently we are incorporating these techniques in reliability analysis tools.

# References


[1] K. Trivedi, J. Dugan, R. Geist, M. Smotherman, B. Rothmann, M. Boyd, and S. Bavuso, "HARP: Introduction and guide for the users," Dept. of Computer Science, Duke University, Durham, NC 27706, March 1986.

[2] R. Sahner, K. Trivedi, and A. Puliafito, *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using SHARPE Software Package,* Kluwer Academic Publishers, 1995.

[3] A. Goyal, W. C. Carter, E. de Douza e Silva, S. S. Lavenberg, and K. S. Trivedi, "The System Availability Estimator," *in Proc. of FTCS-16,* June 1986, pp. 84-89.

[4] R. Butler, "The SURE Reliability Analysis Program," *AIAA Guidance, Navigation, and Control Conference,* Williamsburg, Virginia, August 1986.

[5] J. D. Esary and H. Ziehms, "Reliability Analysis of Phased Missions," *Proc. of the Conf. on Reliability and Fault Tree Analysis,* SIAM, 1975, pp. 213-236.

[6] A. Pedar and V. V. S. Sarma, "Phased-Mission Analysis for Evaluating the Effectiveness of Aerospace Computing-Systems," *IEEE Trans. on Rel.,* vol. R-30, No.5, Dec. 1981, pp. 429-437.

[7] M. Alam and U. Al-Saggaf, "Quantitative Reliability Evaluation of Repairable Phased-Mission Systems Using Markov Approach," *IEEE Trans. on Rel.,* vol. R-35, No.5, Dec. 1986, pp. 498-503.

[8] J. B. Dugan, "Automated Analysis of Phased-Mission Reliability," *IEEE Trans. on Rel.,* vol. R40, No. 1, April 1991, pp. 45-51.

[9] M. Smotherman and K. Zemoudeh, "A Non-Homogeneous Markov Model for Phased-Mission Reliability Analysis," *IEEE Trans. on Rel.,* vol. R-38, No. 5, Dec. 1989, pp. 585-590.

[10] A. Somani, J. Ritcey, and S. Au, "Computationally Efficient Phased-Mission Reliability Analysis for Systems with Variable Configuration," *IEEE Trans. on Rel.,* vol. R-42, Dec. 1992, pp. 504-509.

[11] A. Somani, S. Palnitkar, and T. Sharma, "Reliability Modeling of Systems with Latent Failures Using Markov Chains," *Proc. of RAMS-93,* pp. 120-125.

[12] M. Veeraraghavan and K. S. Trivedi, "An Improved Algorithm for Symbolic Reliability Analysis," *IEEE Trans. on Rel.,* vol. R-40, No. 3, Dec. 1991, pp. 347-358.

[13] A. K. Somani and K. S. Trivedi, "Phased-Mission System Analysis Using Boolean Algebraic Methods," in the Proc. of Sigmetrics, 1994, pp. 98-107

[14] A. K. Somani, "Simplified Phased-Mission System Analysis for Systems with Independent Component Repairs," NASA CR-198318, ICASE Report No. 96-23, March 1996.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | January 1997 | Contractor Report |

**4. TITLE AND SUBTITLE**
BOOLEAN ALGEBRAIC METHODS FOR PHASED-MISSION SYSTEM ANALYSIS

**5. FUNDING NUMBERS**
C NAS1-19480
WU 505-90-52-01

**6. AUTHOR(S)**
Arun K. Somani
Kishor S. Trivedi

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Institute for Computer Applications in Science and Engineering
Mail Stop 403, NASA Langley Research Center
Hampton, VA 23681-0001

**8. PERFORMING ORGANIZATION REPORT NUMBER**
ICASE Report No. 97-3

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
NASA CR-201644
ICASE Report No. 97-3

**11. SUPPLEMENTARY NOTES**
Langley Technical Monitor: Dennis M. Bushnell
Final Report
Submitted to IEEE Transactions on Reliability.

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified–Unlimited

Subject Category 60, 61

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

Most reliability analysis techniques and tools assume that a system is used for a mission consisting of a single phase. However, multiple phases are natural in many missions. The failure rates of components, system configuration, and success (failure) criteria may vary from phase to phase. In addition, the duration of a phase may be deterministic or random. We describe a new technique for phased-mission system reliability analysis based on Boolean algebraic methods. Our technique is computationally efficient and is applicable to a large class of systems for which the failure criterion in each phase can be expressed as a fault tree (or an equivalent representation). Our technique avoids state space explosion that commonly plague Markov chain-based analysis. We develop a phase algebra to account for the effects of variable configurations and failure criteria from phase to phase. Our technique yields exact (as opposed to approximate) results. We demonstrate the use our technique by means of an example and present numerical results to show the effects of mission phases on the system reliability.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| Ultra-Reliable Computer System; Reliability Analysis; Boolean Algebraic Methods; Fault Trees; Phased-Mission Systems; Variable Success (Failure) Criteria; Reconfiguration; Random Phase Duration | 21 |
| | 16. PRICE CODE A03 |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | | |